

What is claimed is:

1. An article comprising a machine readable medium storing instructions that, if executed by a machine, cause the machine to perform a plurality of operations comprising:

- specifying a monitor address;
- suspending a thread until a monitor break event occurs;
- testing whether the monitor break event is a write to the monitor address;
- if the monitor break event is not the write to the monitor address, then suspending the thread again.

2. The article of claim 1 wherein suspending the thread again comprises returning to specifying the monitor address.

3. The article of claim 2 wherein specifying the monitor address comprises executing a MONITOR instruction and wherein suspending the thread until the monitor break event occurs comprises executing an MWAIT instruction.

4. The article of claim 1 wherein said plurality of operations further comprise, after specifying the monitor address and before suspending the thread:

- testing whether data at the monitor address has changed.

5. The article of claim 1 wherein specifying the monitor address comprises executing an

instruction with an operand chosen from a set consisting of a linear address, a virtual address, a physical address, and a relative address.

6. The article of claim 5 wherein the operand is one of a second set consisting of an explicit operand and an implicit operand.

7. The article of claim 1 wherein said monitor address specifies a cache line.

8. The article of claim 2 wherein said plurality of operations further comprise providing a second operand as a mask operand to control which events are monitor break events.

9. An article comprising a machine readable medium storing instructions that, if executed by a machine, cause the machine to perform operations comprising:

- programming a monitor with a monitor address corresponding to a cache line of at least one work location;
- suspending a thread until a monitor break event occurs;
- testing whether the at least one work location indicates a first task is ready to execute;
- testing whether the at least one work location indicates a second task is ready to execute;
- if neither the first task nor the second task is ready to execute, then returning to suspending the thread.

- 1 10. The article of claim 9 wherein returning to suspending the thread until the monitor
2 break event occurs further comprises re-programming the monitor with the monitor
3 address prior to suspending the thread.
1
- 1 11. The article of claim 9 wherein returning to suspending the thread comprises returning
2 to programming the monitor with the monitor address.
1
- 1 12. A method comprising:
2 specifying a monitor address;
3 suspending a thread until a monitor break event occurs;
4 testing whether the monitor break event is a write to the monitor address;
5 if the monitor break event is the write to the monitor address, then suspending the
6 thread again.
1
- 1 13. The method of claim 12 wherein suspending the thread again comprises returning to
2 specifying the monitor address.
1
- 1 14. The method of claim 13 wherein specifying the monitor address comprises executing
2 a MONITOR instruction and wherein suspending the thread until the monitor break
3 event occurs comprises executing an MWAIT instruction.
1
- 1 15. The method of claim 12 wherein said method further comprises, after specifying the
2 monitor address and before suspending the thread:

42390.P13636

1 21. The system of claim 20 wherein said loop comprises:

2 a first instruction to specify the monitor address;

3 a second instruction to suspend said first thread.

1

1 22. The system of claim 21 wherein said loop further comprises a test after said first

2 instruction to determine whether data at the monitor address has changed after

3 execution of the first instruction but before execution of the second instruction,

4 wherein said loop exits without execution of the second instruction if data at the

5 monitor address has changed.

1

1 23. The system of claim 21 wherein said loop further comprises a test after said first

2 instruction to determine whether data at the monitor address has changed after

3 execution of the second instruction wherein said loop performs another iteration if

4 data at the monitor address has not changed.

1

1 24. The system of claim 20 wherein said loop comprises:

2 a test to determine whether a work location in a first cache line indicated by the

3 monitor address contains a first value, wherein a first routine is executed if

4 said work location contains the first value;

5 a second test to determine whether the work location in said first cache line

6 contains a second value, wherein a second routine is executed if said work

7 location contains the second value;

8 an instruction to suspend said first thread if said work location does not contain

9 said first value and said work location does not contain said second value.

1 25. A system comprising:

2 a processor;

3 a monitor;

4 a memory to store an idle loop in a first thread executable by said processor to

5 perform operations comprising:

6 specifying a monitor address;

7 suspending said first thread until a monitor break event occurs;

8 testing whether the monitor break event is a write to the monitor address;

9 if the monitor break event is not the write to the monitor address, then

10 returning to specifying the monitor address.

1

1 26. The system of claim 25 wherein specifying the monitor address comprises executing

2 a first instruction and wherein suspending the thread until the monitor break event

3 occurs comprises executing a second instruction.

1

1 27. The system of claim 25 wherein said operations further comprise, after specifying the

2 monitor address and before suspending the thread:

3 testing whether data at the monitor address has changed.

1

1 28. A method comprising:

2 executing a first instruction in a first thread that specifies a monitor address;

3 executing a second instruction in said first thread to suspend said first thread until

4 a write access implicating said monitor address or an interrupt occurs;
5 executing a plurality of instructions in a second thread;
6 after the write access or the interrupt occurs, testing whether a data element
7 associated with said monitor address has changed;
8 returning to executing the second instruction if the data element has not changed.

1

1 29. The method of claim 28 wherein returning to executing the second instruction
2 comprises returning to executing the first instruction and continuing on to executing
3 the second instruction.

1

1 30. The method of claim 28 further comprising testing whether the data element
2 associated with said monitor address has changed prior to executing said second
3 instruction.